Tango server

A Tango server running on the PiLCs service CPU allows communication with the FPGA and the IO modules attached to it. This page describes the commands provided by the server and their particular purpose

Startup, shutdown, and programming commands

Before using the PiLC it has to be initialized with the *InitPiLC()* command. This command starts the FPGA and configures the IO boards. In order to deactivate the PiLC use the *DeinitPiLC()* command which shuts down the FPGA and the USV.

PiLC initialization and shutdown in Python

```
device.InitPiLC()

# do some useful work here

device.DeInitPiLC()
```

A new program can be loaded by means of the *EPCSProgram(path to file)* command. Its only argument is the path to the **.rbf* file with the new program.

Programming the FPGA from Python

```
file_name = "/home/user/new_program.rbf"
device.EPCSProgram(file_name)
```

As the programming process lasts for about 15 seconds this should not be done from within Jive as it would result in a timout.

FPGA communication

The communication with the FPGA is done by reading and writing to one of its registers. Three Tango commands are available for this purpose

- *ReadFPGA(reg_addr)* – reads data from a single register
- *WriteFPGA([reg_addr,data])* – writes data to a single register (where *data* must be a 32Bit integer value)
- *FPGAReadBurst([first_reg_addr,n_registers])* – read data of *n_registers* starting at the register at address *first_reg_addr*

Please note, that the *WriteFPGA()* function can only be used with the *IO_XX_Data_In_Register* registers. Furthermore, the FPGAReadBurst() command only acts on the subsequent *In* or *Out* registers depending on the register referenced by *first_reg_addr*.

FPGA IO in Python

```
value = device.ReadFPGA(0x01)  #read data from IO_1_Data_In_Register
device.WriteFPGA([0x01,1000])  #write data to IO_1_Data_In_Register
values = device.FPGAReadBurst([0x01,0x02]) #read data from IO_2_Data_Out_Register, IO_3_Data_Out_Register
```

As a Tango command can take only a single argument, multiple arguments must be wrapped in a list. Please note the last line in the previous code snippet: only the *Out* registers are affected by this command as the first register is an *Out* register.

FPGA Register

IO card communication

To read and write data from and to an IO card use

- *ReadIOCard([card_addr,reg_addr])*
- *WriteIOCard([card_addr,reg_addr,value])*

Reading and writing data from and to an IO card

```
value = device.ReadIOCard([0x01,0x01])
device.WriteIOCard([0x,01,0x01,0x03])_
```

IO Cards Register

IO board communication

Data can be read and written from and to the IO board with

- *ReadIOBoard(reg_addr)*
- *WriteIOBoard([reg_addr,value_1,value_2,value_3])*

where *value_1*, *value_2*, and *value_3* are 8Bit values.

Reading and writing data from and to the IO board

```
value = device.ReadIOBoard(0x64)
device.WriteIOBoard([0x01,0xAF,0xAF,0xAF])
```

IO board Register

Display communication

The display board can be accessed via

- *ReadDisplay(reg_addr)*
- *WriteDisplay([reg_addr,value])*

where value is an 8Bit value.

Display Register

Mainboard communication

The mainboard registers can be accessed via

- *ReadMainBoard(reg_addr)*
- *WriteMainBoard([reg_addr,value])*

where *value* is again an 8Bit value.

Reading and writing to the mainboard registers

```
value = device.ReadMainBoard(0x01)
device.WriteMainBoard([0x01,0x01])
```

Mainboard Register